



Conceptos básicos de SQL

NOMBRE Y APELLIDOS: Alex Abades Grimes

El supermercado al por mayor *Pango* quiere disponer de una base de datos para registrar toda la información acerca de los clientes, productos y pedidos que se han registrado como parte de sus operaciones.

Consideraciones para la entrega y realización de la PEC:

- Todo lo que se pide en esta PEC está explicado en los bloques didácticos 1 y 2.
- Se recomienda la utilización de **pgAdmin** para la implementación de toda la PEC.
- Cada respuesta a los ejercicios ha de entregarse en un fichero *.sql* diferente, con el nombre indicado.
- Las capturas de pantalla de los ejercicios (y explicaciones pertinentes) han de proporcionarse en este documento plantilla.
- Se debe de realizar la entrega de todos los ficheros de la PEC (tanto los ficheros *.sql* como el documento con explicaciones y capturas de pantalla) en un único fichero comprimido *.zip*.

Consideraciones para la evaluación del ejercicio:

- Se tendrá muy en cuenta el uso de la convención de nombres, así como la aplicación de las buenas prácticas de codificación en SQL y de consultas en el código que creéis, proporcionadas en el aula. Es decir: código con sangrado, uso de cláusulas SQL de forma correcta, comentarios, etc.
- Los *scripts* proporcionados por el estudiante con las soluciones de los ejercicios han de ejecutarse correctamente.
- **Importante:** Las sentencias SQL proporcionadas en los *scripts* han de ser creadas de forma manual y no mediante asistentes que PostgreSQL/pgAdmin puedan proporcionar. Se pretende aprender SQL y no la utilización de asistentes.
- Las sentencias SQL proporcionadas en los ejercicios han de ser **solamente** aquellas que pide el enunciado y ninguna otra más.
- Las sentencias SQL deben **escribirse usando la información proporcionada en el enunciado, y no otra información** que se pueda deducir de los datos.



Contenido

Índice de ilustraciones	3
EJERCICIO 1 (20%)	4
EJERCICIO 2 (35%)	7
Apartado a)	7
Apartado b)	7
Apartado c)	8
Apartado d)	8
Apartado e)	8
EJERCICIO 3 (20%)	10
Apartado 1	10
Apartado 2	11
Apartado 3	11
Apartado 4	12
EJERCICIO 4 (25%)	14
Apartado 1)	14
Apartado 2)	15
Apartado 3)	16



Índice de ilustraciones

Ilustración 1. Ejercicio 1: Menú pgAdmin.....	4
Ilustración 2. Ejercicio 1: tb_category	4
Ilustración 3. Ejercicio 1: tb_client	5
Ilustración 4. Ejercicio 1: tb_order.....	5
Ilustración 5. Ejercicio 1: tb_order_line.....	5
Ilustración 6. Ejercicio 1: tb_product.....	5
Ilustración 7. Ejercicio 1: tb_subcategory	6
Ilustración 8. Ejercicio 2: Apartado a).....	7
Ilustración 9. Ejercicio 2: Apartado b)	7
Ilustración 10. Ejercicio 2: Apartado c).....	8
Ilustración 11 Ejercicio 2: Apartado d)	8
Ilustración 12 Ejercicio 2: Apartado d)	9
Ilustración 13. Ejercicio 3: Apartado 1)	11
Ilustración 14. Ejercicio 3: Apartado 2)	11
Ilustración 15. Ejercicio 3: Apartado 3)	12
Ilustración 16. Ejercicio 3: Apartado 4)	13
Ilustración 17. Ejercicio 4: Apartado 1) Sección a).....	14
Ilustración 18. Ejercicio 4: Apartado 1) Sección b)	15
Ilustración 19. Ejercicio 4: Apartado 2)	16



EJERCICIO 1 (20%)

Se pide proporcionar las sentencias SQL para crear los componentes especificados a continuación. El resultado de este ejercicio ha de entregarse en un fichero llamado **pec1_ej1.sql**.

Una vez ejecutadas las sentencias SQL de creación de las estructuras que se requieren, se pide ejecutar el script **BBDD_Ventas (datos).sql** adjunto. **Este fichero NO puede ser modificado de ningún modo para la realización de la PEC**, ni debe copiarse como parte de ningún otro fichero (aunque no se cambie ninguna de sus sentencias). Es necesario que todas las sentencias SQL de este fichero se ejecuten correctamente, tal y como se proporciona.

A continuación, se muestran las tablas creadas. Estas contienen ya los datos y modificaciones pertinentes, debido a que no ha sido hasta finalizar la práctica que he pensado en poner fotografías de las tablas.

A continuación, se muestra una ilustración del menú desplegable del pgAdmin.

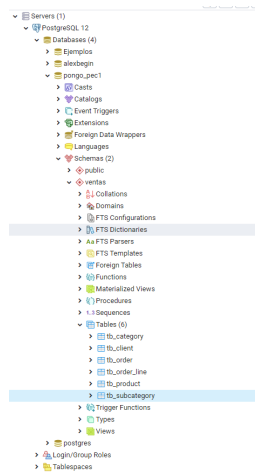


Ilustración 1. Ejercicio 1: Menú pgAdmin.

Tabla de categoría **tb_category**:

category_code	category_name	created_by_user	created_date	updated_date	vat
[PK] character (5)	character varying (70)	character varying (10)	date	date	numeric (4,2)
1 CT001	Frutas	OS_SYSTEM	[null]	[null]	0.08
2 CT002	Bebidas	OS_SYSTEM	[null]	[null]	0.08
3 CT003	Congelados	OS_SYSTEM	[null]	[null]	0.08
4 CT004	Lácteos	OS_SYSTEM	[null]	[null]	0.08
5 CT005	Carnes	OS_SYSTEM	[null]	[null]	0.08
6 CT006	Higiene	OS_SYSTEM	[null]	[null]	0.08

Ilustración 2. Ejercicio 1: **tb_category**



Tabla de clientes tb_client:

Data Output	Explain	Messages	Notifications
client_code [PK] character (5)	client_name character varying (40)	address character varying (140)	city character varying (25)
1 C0001	Eroski	Las Rozas 23	Madrid
2 C0002	Gadis	Polígono Pocomaco 21	A Coruña
3 C0003	Mercadona	Av. Castellana 35	Valencia
4 C0004	Eroski - Madrid	[null]	Madrid
5 C0005	Eroski - Barcelona	[null]	Barcelona
6 C0006	Eroski - Santiago	[null]	Santiago de Compostela
7 C0007	Gadis - País Vasco	C/Estación	Bilbao
8 C0008	Gadis - Galicia	As Pontes	Ourense
9 C0009	Mercadona - Madrid	Chueca 1	Madrid
10 C0010	Mercadona - Barcelona	Av. Layetana	Barcelona
11 C0011	Mercadona - Cracovia	ul. Grodzka 15	Cracovia
12 C0012	Mercadona - Varsovia	ul. Karmelica 15	Varsovia
13 C0013	Eroski - Getafe	Razados 23	Getafe
14 C0014	Eroski - Valencias	Av. Victoria 89	Valencias
15 C0015	Eroski - Barcelona	[null]	Barcelona
16 C0016	Gadis - Bilbao	[null]	Bilbao
17 C0017	Gadis - San Sebastián	As Pontes	Ourense
18 C0018	Gadis - Santiago	[null]	Santiago de Compostela

Ilustración 3. Ejercicio 1: tb_client

Tabla de pedidos tb_order:

Data Output	Explain	Messages	Notifications
order_number [PK] character (10)	client_code character (10)	order_date date	delivery_date date
1 ON2016#001	C0013	2016-01-05	2016-01-06
2 ON2016#002	C0013	2016-01-07	2016-01-13
3 ON2016#003	C0015	2016-01-15	2016-01-16
4 ON2016#004	C0015	2016-01-17	2016-01-21
5 ON2016#005	C0016	2016-01-01	2016-01-03
6 ON2016#006	C0017	2016-01-05	2016-01-07
7 ON2016#007	C0017	2016-01-06	2016-01-15

Ilustración 4. Ejercicio 1: tb_order

Tabla de pedidos abiertos tb_order_line:

Data Output	Explain	Messages	Notifications
order_number [PK] character (10)	order_line_number [PK] integer	product_code character (5)	quantity integer
1 ON2016#001	1	P0001	12
2 ON2016#001	2	P0003	5
3 ON2016#001	3	P0007	10
4 ON2016#002	1	P0010	3
5 ON2016#002	2	P0002	19
6 ON2016#003	1	P0009	7
7 ON2016#005	1	P0018	10

Ilustración 5. Ejercicio 1: tb_order_line

Tabla de pedidos abiertos tb_product:

Data Output	Explain	Messages	Notifications
product_code [PK] character (5)	subcategory_code character (5)	product_name character varying (60)	price numeric (14,2)
1 P0001	SC001	Naranjas España	1.99
2 P0002	SC001	Limones España	1.79
3 P0003	SC001	Pomelos Grecias	0.98
4 P0004	SC002	Manzana Golden	1.48
5 P0005	SC002	Manzana Royal	1.63
6 P0006	SC003	Mejón Grecia	3.39
7 P0007	SC004	Bezoya	0.39

Ilustración 6. Ejercicio 1: tb_product



Tabla de pedidos abiertos tb_product:

	Data Output	Explain	Messages	Notifications				
	subcategory_code [PK] character (5)	category_code character (5)	subcategory_name character varying (55)	created_by_user character varying (10)	created_date date	updated_date date		
1	SC001	CT001	Cítricos	OS_SYSTEM	[null]	[null]		
2	SC002	CT001	Manzanas y peras	OS_SYSTEM	[null]	[null]		
3	SC003	CT001	Melón y sandía	OS_SYSTEM	[null]	[null]		
4	SC004	CT002	Aguas	OS_SYSTEM	[null]	[null]		
5	SC005	CT002	Gaseosas	OS_SYSTEM	[null]	[null]		
6	SC007	CT003	Pizzas	OS_SYSTEM	[null]	[null]		
7	SC008	CT003	Verduras	OS_SYSTEM	[null]	[null]		

Ilustración 7. Ejercicio 1: tb_subcategory



EJERCICIO 2 (35%)

Se pide proporcionar, para cada apartado, una única consulta en lenguaje SQL, y el resultado de dicha consulta mediante una captura del resultado en pgAdmin.

Las consultas SQL de este ejercicio han de entregarse en un fichero llamado **pec1_ej2.sql** (las capturas de pantalla, y explicaciones si son necesarias, han de entregarse en un documento adjunto diferente, en formato PDF obligatoriamente).

Apartado a)

Obtener un listado de los países en los cuales la empresa tiene algún cliente (independientemente de si en la base de datos consta o no algún pedido de este cliente), así como el número total de clientes que tiene de cada país. El resultado debe de estar ordenado descendientemente por el número de clientes calculado.

A continuación, se muestra una ilustración del resultado de la query.

	Data Output	Explain	Messages	Notificator
	<div>country</div> <div>character varying (60)</div>		<div>num_clients</div> <div>bigint</div>	
1	Spain			20
2	Poland			4
3	France			1

Ilustración 8. Ejercicio 2: Apartado a)

Apartado b)

La empresa desea analizar las ciudades desde las que tiene mayor cantidad de pedidos. Por ello se desea obtener el listado de ciudades desde las que se ha realizado algún pedido, así como el número de pedidos realizados desde esa ciudad. Mostrar el nombre de la ciudad, así como el total de pedidos realizados, ordenados descendientemente por el número de pedidos, y ascendentemente por nombre de la ciudad. Se desea obtener, únicamente, aquellas ciudades desde las que se han realizado, como mínimo 3 pedidos.

A continuación, se muestra una ilustración del resultado de la query.

	Data Output	Explain	Messages	Notifications
	<div>num_pedidos</div> <div>bigint</div>		<div>country</div> <div>character varying (60)</div>	
1	20	Spain		

Ilustración 9. Ejercicio 2: Apartado b)



Apartado c)

Obtener el nombre de todos aquellos productos que pertenecen a la categoría **Bebidas**, y cuyo precio está por debajo de 0.80€.

A continuación, se muestra una ilustración del resultado de la query.

	subcategory_code character (5)	category_code character (5)	category_name character varying (70)	price numeric (14,2)
1	SC004	CT002	Bebidas	0.39
2	SC004	CT002	Bebidas	0.59
3	SC004	CT002	Bebidas	0.19

Ilustración 10. Ejercicio 2: Apartado c)

Apartado d)

Obtener el código y el nombre de todas las categorías para las que no consta que se haya definido ninguna subcategoría.

A continuación, se muestra una ilustración del resultado de la query.

	category_code [PK] character (5)	category_name character varying (70)
1	CT006	Higiene

Ilustración 11 Ejercicio 2: Apartado d)

Apartado e)

Para cada pedido, se desea calcular el valor total del mismo, teniendo en cuenta todas sus líneas de pedido, la cantidad de producto en cada línea, y el precio unitario que consta en esa línea. Debe mostrarse el número de pedido, nombre del cliente, la fecha de pedido y la fecha de entrega (*delivery_date*), así como el valor total del pedido. Únicamente interesan los pedidos realizados con posterioridad al 5 de enero del 2016, y que tengan un valor total superior a 100€. El resultado debe ordenarse primero por nombre de cliente, después por fecha de pedido, y finalmente por fecha de entrega, todos de forma ascendente.

A continuación, se muestra una ilustración del resultado de la query.



Data Output

Explain

Messages

Notifications

	<div>client_name</div> <div>character varying (40)</div>	<div>order_number</div> <div>character (10)</div>	<div>order_date</div> <div>date</div>	<div>delivery_date</div> <div>date</div>	<div>total_amout_order</div> <div>numeric</div>
1	Gadis - San Sebastián	ON2016#007	2016-01-06	2016-01-15	105.57
2	Mercadona - Algeciras	ON2016#016	2016-01-12	2016-01-17	125.67
3	Mercadona - Vallecas	ON2016#012	2016-01-16	2016-01-19	106.83

Ilustración 12 Ejercicio 2: Apartado d)



EJERCICIO 3 (20%)

Se pide proporcionar, para cada apartado, **la sentencia o conjunto de sentencias en lenguaje SQL, y el resultado de dichas sentencias mediante una captura del resultado**. Para poder realizar este ejercicio, el ejercicio 1 ha de estar finalizado.

Estas sentencias deben **escribirse usando la información proporcionada en el enunciado, y no otra información** que se pueda deducir de los datos. Por ejemplo, si el enunciado hace referencia a la categoría 'Bebidas', es esta la información que debe de aparecer en la consulta, y no el código correspondiente 'CT002'.

Las sentencias SQL de este ejercicio han de entregarse en un fichero llamado **pec1_ej3.sql** (las capturas de pantalla han de entregarse en un documento adjunto, el mismo que se haya usado para las capturas de pantalla del Ejercicio 2, en formato PDF).

Apartado 1

Deben realizarse diversos cambios de precios en la tabla '**tb_product**'.

- a) El producto llamado '**El Coto**' ha subido su precio considerablemente, un 25%.
- b) Las '**Manzanas y peras**' deben bajar su precio un 7%.
- c) Por último, todos los '**Lácteos**' aumentan su precio un 9%.

A continuación, se muestra una ilustración del resultado de la query subrayado en amarillo. Esta query solo debe ejecutarse una vez, tal y como verán en el archivo .sql, el UPDATE va en función de la propia columna en sí. Variable dependiente. Motivo por el cuál, si se ejecuta más de una vez el valor que obtendrán los productos deseados será mayor o menor del esperado.

$a = a * 2$

Si esta misma consulta se ejecuta más de una vez, habiendo fijado $a = 1$, el valor incrementara por un factor de 2 cada vez.



Data Output	Explain	Messages	Notifications			
product_code (PK) character (5)	subcategory_code character (5)	product_name character varying (60)	price numeric (14,2)	created_by_user character varying (10)	created_date date	update_date date
1 P0001	SC001	Naranjas España	1.99	OS_SYSTEM	[null]	[null]
2 P0002	SC001	Limonas España	1.79	OS_SYSTEM	[null]	[null]
3 P0003	SC001	Pomelos Grecias	0.98	OS_SYSTEM	[null]	[null]
4 P0004	SC002	Mancana Golden	1.48	OS_SYSTEM	[null]	[null]
5 P0005	SC002	Mancana Royal	1.63	OS_SYSTEM	[null]	[null]
6 P0006	SC003	Melón Grecia	3.39	OS_SYSTEM	[null]	[null]
7 P0007	SC004	Bezoya	0.39	OS_SYSTEM	[null]	[null]
8 P0008	SC004	Cabreola	0.59	OS_SYSTEM	[null]	[null]
9 P0009	SC004	Del grifo	0.19	OS_SYSTEM	[null]	[null]
10 P0010	SC005	Lapitusa	1.39	OS_SYSTEM	[null]	[null]
11 P0011	SC013	El Coto	4.61	OS_SYSTEM	[null]	[null]
12 P0012	SC013	Azplicueta	5.89	OS_SYSTEM	[null]	[null]
13 P0013	SC007	Pizza Margarita	2.69	OS_SYSTEM	[null]	[null]
14 P0014	SC007	Pizza Cuatro Quesos	2.89	OS_SYSTEM	[null]	[null]
15 P0015	SC008	Guisantes El Tenedor	1.29	OS_SYSTEM	[null]	[null]
16 P0016	SC008	Brocoli El Tenedor	2.29	OS_SYSTEM	[null]	[null]
17 P0017	SC014	Yogur Yogurt	0.43	OS_SYSTEM	[null]	[null]
18 P0018	SC009	Leche desnatada	0.64	OS_SYSTEM	[null]	[null]
19 P0019	SC009	Leche entera	0.85	OS_SYSTEM	[null]	[null]
20 P0020	SC009	Leche semi-desnatada	0.75	OS_SYSTEM	[null]	[null]
21 P0021	SC010	Alitas pollo	1.39	OS_SYSTEM	[null]	[null]
22 P0022	SC010	Pollo entero	4.39	OS_SYSTEM	[null]	[null]
23 P0023	SC012	Pata de ternera	3.59	OS_SYSTEM	[null]	[null]
24 P0024	SC012	Faldá de ternera	2.99	OS_SYSTEM	[null]	[null]
25 P0025	SC014	Danio	1.08	OS_SYSTEM	[null]	[null]
26 P0026	SC015	Papel	3.59	OS_SYSTEM	[null]	[null]

Ilustración 13. Ejercicio 3: Apartado 1)

Apartado 2

Para simplificar el contenido de la base de datos, se nos pide que se eliminen todas aquellas subcategorías para las cuales no hay ningún producto en la base de datos que pertenezca a dichas subcategorías.

A continuación, se muestra una ilustración del resultado de la query.

25 LEFT JOIN tb_product

26 ON sct.subcategory_code = tb_product.subcategory_code

27 WHERE tb_product.product_code IS NULL

Data Output

Explain

Messages

Notifications

<div><div>▲</div></div> subcategory_code	subcategory_name	product_code	subcategory_code	product_name	price	
character (5)	character varying (55)	character (5)	character (5)	character varying (60)	numeric (14,2)	
1 SC006	Zumos	[null]	[null]	[null]	[null]	
2 SC011	Pavo	[null]	[null]	[null]	[null]	

Ilustración 14. Ejercicio 3: Apartado 2)

Apartado 3

Nos dicen que falta añadir información en la tabla **tb_client** para completar la jerarquía existente entre algunos clientes, que no se ha almacenado todavía.

- a) Cliente con código **C0030**, con nombre **Lidl**, dirección **Drungüerkanster 38**, ciudad **Frankfurt**, país **Germany**, teléfono no proporcionado, y email **info@lidl.de**.



- b) Cliente con código **C0031**, con nombre **Lidl - Frankfurt**, dirección **Drungüerkanster 128**, ciudad **Frankfurt**, país **Germany**, teléfono **998858896**, y email **frankfurt@lidl.de**. Este cliente tiene como padre el cliente **Lidl**. Para esta sentencia no puede usarse explícitamente el código 'C0030' de Lidl.

Se pide que se registre toda esta información con usuario creador '**Alumno-UOC**' y fecha de creación **31 Octubre 2020**.

A continuación, se muestra una ilustración del resultado de la query, los últimos registros.

Data Output	client_code	client_name	address	city	country	contact_email	phone	parent_client_code	created_by_user	created_date	update_date
	[PK] character (5)	character varying (40)	character varying (140)	character varying (25)	character varying (60)	character varying (100)	character varying (15)	character (5)	character varying (10)	date	date
1	C0001	Eroski	Las Rozas 23	Madrid	Spain	[null]	[null]	[null]	OS_SYSTEM	[null]	[null]
2	C0002	Gadis	Poligono Pocomaco 21	A Coruña	Spain	[null]	+34537789995	[null]	OS_SYSTEM	[null]	[null]
3	C0003	Mercadona	Av. Castellana 35	Valencia	Spain	info@mercadona.es	[null]	[null]	OS_SYSTEM	[null]	[null]
4	C0004	Eroski - Madrid	[null]	Madrid	Spain	[null]	[null]	C0001	apemag	[null]	[null]
5	C0005	Eroski - Barcelona	[null]	Barcelona	Spain	info@eroski.es	[null]	C0001	apemag	[null]	[null]
6	C0006	Eroski - Santiago	[null]	Santiago de Compostela	Spain	[null]	981 236 698	C0001	apemag	[null]	[null]
7	C0007	Gadis - País Vasco	C/Estación	Bilbao	Spain	[null]	[null]	C0002	asavar	[null]	[null]
8	C0008	Gadis - Galicia	As Pontes	Ourense	Spain	info@eroski.es	[null]	C0002	asavar	[null]	[null]
9	C0009	Mercadona - Madrid	Chueca 1	Madrid	Spain	[null]	[null]	C0003	asavar	[null]	[null]
10	C0010	Mercadona - Barcelona	Av. Layetana	Barcelona	Spain	info@mercadona.es	[null]	C0003	apemag	[null]	[null]
11	C0011	Mercadona - Cracovia	ul. Grodzka 15	Cracovia	Poland	info@mercadona.es	+48888885236	C0003	apemag	[null]	[null]
12	C0012	Mercadona - Varsovia	ul. Karmelica 15	Varsovia	Poland	info@mercadona.es	+48888881136	C0003	apemag	[null]	[null]
13	C0013	Eroski - Getafe	Razados 23	Getafe	Spain	info@eroski.es	[null]	C0004	apemag	[null]	[null]
14	C0014	Eroski - Valencias	Av. Victoria 89	Valencias	Spain	info@eroski.es	[null]	C0004	apemag	[null]	[null]
15	C0015	Eroski - Barcelona	[null]	Barcelona	Spain	[null]	981 236 698	C0005	apemag	[null]	[null]
16	C0016	Gadis - Bilbao	[null]	Bilbao	Spain	[null]	[null]	C0007	asavar	[null]	[null]
17	C0017	Gadis - San Sebastián	As Pontes	Ourense	Spain	info@eroski.es	[null]	C0007	asavar	[null]	[null]
18	C0018	Gadis - Santiago	[null]	Santiago de Compostela	Spain	[null]	[null]	C0008	asavar	[null]	[null]
19	C0019	Gadis - Lugo	[null]	Lugo	Spain	info@eroski.es	[null]	C0008	asavar	[null]	[null]
20	C0020	Mercadona - Getafe	[null]	Getafe	Spain	[null]	[null]	C0009	asavar	[null]	[null]
21	C0021	Mercadona - Valencias	Av. Simples 112	Valencias	Spain	info@mercadona.es	[null]	C0009	apemag	[null]	[null]
22	C0022	Mercadona - Algeciras	ul. Grodzka 15	Algeciras	Spain	info@mercadona.es	[null]	C0009	apemag	[null]	[null]
23	C0023	Mercadona - Cracovia	ul. Grodzka 15	Cracovia	Poland	info@mercadona.es	+48888881136	C0011	apemag	[null]	[null]
24	C0024	Mercadona - Varsovia	ul. Karmelica 15	Varsovia	Poland	info@mercadona.es	+48888881136	C0012	apemag	[null]	[null]
25	C0026	Dia	Camps Elisees 125	Paris	France	dia@dia.fr	[null]	[null]	OS_SYSTEM	[null]	[null]
26	C0030	Lidl	Drungierkanster 38	Frankfurt	Germany	info@lidl.de	[null]	[null]	Alumno-UOC	2020-10-31	[null]
27	C0031	Lidl - Frankfurt	Drungierkanster 128	Frankfurt	Germany	frankfurt@lidl.de	998 858 896	C0031	Alumno-UOC	2020-10-31	[null]

Ilustración 15. Ejercicio 3: Apartado 3)

Apartado 4

Es necesario corregir la base de datos, de forma que aquellas subcategorías que no tienen categoría asignada, sí la tengan. Concretamente, hay que proporcionar cuatro sentencias UPDATE diferentes:

- 'Vinos', son una subcategoría de 'Bebidas'.
- 'Yogures', son 'Lácteos'.
- 'Tertera', son un tipo de 'Carnes'.
- 'Papel', podemos considerarlo como perteneciente a la categoría 'Higiene'.

A continuación, se muestra una ilustración del resultado de la query, marcado en amarillo.



	Data Output	Explain	Messages	Notifications				
	subcategory_code [PK] character (5)		category_code character (5)	subcategory_name character varying (55)	created_by_user character varying (10)	created_date date	updated_date date	
1	SC001		CT001	Cítricos	OS_SYSTEM	[null]	[null]	
2	SC002		CT001	Manzanas y peras	OS_SYSTEM	[null]	[null]	
3	SC003		CT001	Melón y sandía	OS_SYSTEM	[null]	[null]	
4	SC004		CT002	Aguas	OS_SYSTEM	[null]	[null]	
5	SC005		CT002	Gaseosas	OS_SYSTEM	[null]	[null]	
6	SC007		CT003	Pizzas	OS_SYSTEM	[null]	[null]	
7	SC008		CT003	Verduras	OS_SYSTEM	[null]	[null]	
8	SC009		CT004	Leche	OS_SYSTEM	[null]	[null]	
9	SC010		CT005	Pollo	OS_SYSTEM	[null]	[null]	
10	SC012		CT005	Ternera	OS_SYSTEM	[null]	[null]	
11	SC013		CT002	Vinos	OS_SYSTEM	[null]	[null]	
12	SC014		CT004	Yogures	OS_SYSTEM	[null]	[null]	
13	SC015		CT006	Papel	OS_SYSTEM	[null]	[null]	

Ilustración 16. Ejercicio 3: Apartado 4)



EJERCICIO 4 (25%)

Se pide proporcionar, para cada apartado, la **sentencia o conjunto de sentencias en lenguaje SQL**, y el **resultado de dichas sentencias mediante una captura del resultado**.

Las sentencias SQL de este ejercicio han de entregarse en un fichero llamado **pec1_ej4.sql** (las capturas de pantalla han de entregarse en un documento adjunto, el mismo que se haya usado para las capturas de pantalla de los ejercicios anteriores, en formato PDF).

Apartado 1)

Se desea modificar la base de datos para que permita representar el IVA que debe ser aplicado al precio de los productos.

- a) Para empezar, se necesita añadir una columna a la categoría de los productos (**tb_category**), que se llame **vat** (de 'value added tax'). Debe permitir **2 dígitos no decimales, y 2 decimales**, y **no** debe permitir **valores nulos**. Para simplificar la situación, supongamos que **por defecto** todas las categorías tendrán, inicialmente, un mismo IVA del **8%**: asignar el valor 0.08 a esta columna para todas las categorías.

A continuación, se muestra una ilustración del resultado de la query.







Data Output		Explain	Messages	Notifications		
	 category_code [PK] character (5)	 category_name character varying (70)	 created_by_user character varying (10)	 created_date date	 updated_date date	 vat numeric (4,2)
1	CT001	Frutas	OS_SYSTEM	[null]	[null]	0.08
2	CT002	Bebidas	OS_SYSTEM	[null]	[null]	0.08
3	CT003	Congelados	OS_SYSTEM	[null]	[null]	0.08
4	CT004	Lácteos	OS_SYSTEM	[null]	[null]	0.08
5	CT005	Carnes	OS_SYSTEM	[null]	[null]	0.08
6	CT006	Higiene	OS_SYSTEM	[null]	[null]	0.08

Ilustración 17. Ejercicio 4: Apartado 1) Sección a)

- b) A continuación, es necesario añadir en la tabla de líneas de pedido una nueva columna llamada **total_vat_sum** (suma total IVA), con tipo de dato numérico que permita **20 dígitos no decimales y 4 dígitos decimales**, y **no** debe permitir **valores nulos**. La columna ha de actualizarse para todas las líneas de pedido con el valor obtenido de multiplicar la cantidad de unidades por el precio unitario, añadiéndole el IVA almacenado en la categoría correspondiente (que podría, o no, ser siempre del 8%).

A continuación, se muestra una ilustración del resultado de la query.



Data Output		Explain	Messages	Notifications					
	order_number [PK] character (10)	order_line_number [FK] integer	product_code character (5)	quantity integer	unit_price numeric (14,2)	created_by_user character varying (10)	created_date date	updated_date date	total_vat_sum numeric (20,4)
1	ON2016#001		1 P0001	12	1.99	OS_SYSTEM	[null]	[null]	25.7904
2	ON2016#001		2 P0003	5	0.79	OS_SYSTEM	[null]	[null]	4.2660
3	ON2016#001		3 P0007	10	0.39	OS_SYSTEM	[null]	[null]	4.2120
4	ON2016#002		1 P0010	3	1.39	OS_SYSTEM	[null]	[null]	4.5036
5	ON2016#002		2 P0002	19	1.79	OS_SYSTEM	[null]	[null]	36.7308
6	ON2016#003		1 P0009	7	0.19	OS_SYSTEM	[null]	[null]	1.4364
7	ON2016#005		1 P0018	10	0.59	OS_SYSTEM	[null]	[null]	6.3720
8	ON2016#005		2 P0021	17	1.39	OS_SYSTEM	[null]	[null]	25.5204
9	ON2016#006		1 P0024	20	2.39	OS_SYSTEM	[null]	[null]	51.6240
10	ON2016#006		2 P0003	2	0.79	OS_SYSTEM	[null]	[null]	1.7064
11	ON2016#006		3 P0010	13	1.39	OS_SYSTEM	[null]	[null]	19.5156
12	ON2016#006		4 P0024	17	2.39	OS_SYSTEM	[null]	[null]	43.8804
13	ON2016#007		1 P0001	16	1.99	OS_SYSTEM	[null]	[null]	34.3872
14	ON2016#007		2 P0005	17	1.75	OS_SYSTEM	[null]	[null]	32.1300
15	ON2016#007		3 P0016	18	2.29	OS_SYSTEM	[null]	[null]	44.5176
16	ON2016#007		4 P0020	4	0.69	OS_SYSTEM	[null]	[null]	2.9808
17	ON2016#008		1 P0006	3	3.59	OS_SYSTEM	[null]	[null]	11.6316
18	ON2016#008		2 P0005	20	1.75	OS_SYSTEM	[null]	[null]	37.8000
19	ON2016#010		1 P0024	14	2.39	OS_SYSTEM	[null]	[null]	36.1368
20	ON2016#010		2 P0008	20	0.59	OS_SYSTEM	[null]	[null]	12.7440
21	ON2016#010		3 P0009	19	0.19	OS_SYSTEM	[null]	[null]	3.8988
22	ON2016#012		1 P0024	14	2.39	OS_SYSTEM	[null]	[null]	36.1368
23	ON2016#012		2 P0003	14	0.79	OS_SYSTEM	[null]	[null]	11.9448
24	ON2016#012		3 P0002	5	1.79	OS_SYSTEM	[null]	[null]	9.6660
25	ON2016#012		4 P0001	10	1.99	OS_SYSTEM	[null]	[null]	21.4920
26	ON2016#012		5 P0024	14	2.39	OS_SYSTEM	[null]	[null]	36.1368
27	ON2016#013		1 P0003	14	0.79	OS_SYSTEM	[null]	[null]	11.9448
28	ON2016#014		1 P0001	10	1.99	OS_SYSTEM	[null]	[null]	21.4920
29	ON2016#014		2 P0024	1	2.39	OS_SYSTEM	[null]	[null]	2.5812

Ilustración 18. Ejercicio 4: Apartado 1) Sección b)

Apartado 2)

Nos comentan que la columna **reception_date** en la tabla de pedidos no se utiliza. Debe eliminarse.

A continuación, se muestra una ilustración del resultado de la query.



	Data Output	Explain	Messages	Notifications					
	order_number [PK] character (10)	client_code character (10)	order_date date	delivery_date date	created_by_user character varying (10)	created_date date	updated_date date		
1	ON2016#001	C0013	2016-01-05	2016-01-06	OS_SYSTEM	[null]	[null]		
2	ON2016#002	C0013	2016-01-07	2016-01-13	OS_SYSTEM	[null]	[null]		
3	ON2016#003	C0015	2016-01-15	2016-01-16	OS_SYSTEM	[null]	[null]		
4	ON2016#004	C0015	2016-01-17	2016-01-21	OS_SYSTEM	[null]	[null]		
5	ON2016#005	C0016	2016-01-01	2016-01-03	OS_SYSTEM	[null]	[null]		
6	ON2016#006	C0017	2016-01-05	2016-01-07	OS_SYSTEM	[null]	[null]		
7	ON2016#007	C0017	2016-01-06	2016-01-15	OS_SYSTEM	[null]	[null]		
8	ON2016#008	C0018	2016-01-08	2016-01-14	OS_SYSTEM	[null]	[null]		
9	ON2016#009	C0018	2016-01-25	2016-01-27	OS_SYSTEM	[null]	[null]		
10	ON2016#010	C0018	2016-01-22	2016-01-29	OS_SYSTEM	[null]	[null]		
11	ON2016#011	C0020	2016-01-14	2016-01-15	OS_SYSTEM	[null]	[null]		
12	ON2016#012	C0021	2016-01-16	2016-01-19	OS_SYSTEM	[null]	[null]		
13	ON2016#013	C0021	2016-01-31	2016-01-31	OS_SYSTEM	[null]	[null]		
14	ON2016#014	C0021	2016-01-21	2016-01-25	OS_SYSTEM	[null]	[null]		
15	ON2016#015	C0022	2016-01-13	2016-01-17	OS_SYSTEM	[null]	[null]		
16	ON2016#016	C0022	2016-01-12	2016-01-17	OS_SYSTEM	[null]	[null]		
17	ON2016#017	C0022	2016-01-14	2016-01-26	OS_SYSTEM	[null]	[null]		
18	ON2016#018	C0024	2016-01-14	2016-01-19	OS_SYSTEM	[null]	[null]		
19	ON2016#019	C0024	2016-01-19	2016-01-30	OS_SYSTEM	[null]	[null]		
20	ON2016#020	C0024	2016-01-20	2016-01-28	OS_SYSTEM	[null]	[null]		
21	ON2016#021	C0015	2016-01-17	2016-01-22	OS_SYSTEM	[null]	[null]		
22	ON2016#022	C0018	2016-01-17	2016-01-22	OS_SYSTEM	[null]	[null]		
23	ON2016#023	C0018	2016-01-17	2016-01-24	OS_SYSTEM	[null]	[null]		

Ilustración 19. Ejercicio 4: Apartado 2)

Apartado 3)

Nos comentan también que el número de líneas de pedido máximas que podría solicitar un cliente es de 35. Se pide modificar la restricción existente en la tabla de líneas de pedido para implementar este requisito.



Criterios de valoración

En el enunciado se indica el peso/valoración de cada ejercicio.

Para conseguir la puntuación máxima en los ejercicios, es necesario explicar con claridad la solución que se propone.

Formato y fecha de entrega

Tenéis que enviar un único fichero comprimido `.zip` con todos vuestros documentos (tanto los ficheros `.sql` como el documento con explicaciones y capturas de pantalla) al buzón de Entrega y registro de EC disponible en el aula (apartado Evaluación).

Recordad que:

1. La respuesta a los ejercicios deben darse cada uno en un fichero `.sql` diferente, según se indica en cada ejercicio, con el nombre correspondiente.
2. Las capturas de pantalla de los ejercicios (y explicaciones pertinentes) se deben incluir en un documento aparte **en formato pdf** (tenéis disponible una plantilla para ello, **indicad vuestro nombre en el documento**, por favor). **No se debe** entregar un documento en formato Word o similar (`.docx`, `.doc`, `.odt`).
3. El nombre del fichero debe contener el código de la asignatura, vuestro apellido y vuestro nombre, así como el número de actividad. Es decir, el nombre del fichero debe tener **exactamente** la siguiente forma: **B0472_Apellido1_Apellido2_Nombre_PEC1.zip** (poniendo el nombre y apellido/s que corresponda).

La fecha límite para entregar la PEC es el **6/12/2020**.

Nota: Propiedad intelectual

Al presentar una práctica o PEC que haga uso de recursos ajenos, se tiene que presentar junto con ella un documento en que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y su estatus legal: si la obra está protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL etc.). El estudiante tendrá que asegurarse que la licencia que sea no impide específicamente su uso en el marco de la práctica o PEC. En caso de no encontrar la información correspondiente tendrá que asumir que la obra está protegida por el copyright.

Será necesario, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente, si así corresponde.